unpatentable over Leach et al. in view of Hailpern et al. and further in view of Hughes (U.S. Patent No. 6,345,382).

Based on the following arguments, Applicants respectfully traverse the rejection of claims 1, 2, 4-7, and 9-13 under 35 U.S.C. § 102(e) and/or 35 U.S.C. § 103(a).

Applicants traverse the rejection of claims 1-4, 6-9, and 12 under 35 U.S.C. § 102(e) because Leach et al. does not teach each and every step and/or element of these claims.

Leach et al. discloses a method and system for aggregating objects within an enclosed object. The method and system allows for static and dynamic aggregation. In static aggregation, an interface of an enclosing object knows in advance (e.g., before runtime) how to return an identifier to an external interface of an enclosed object. In dynamic aggregation, an enclosed object is added to the enclosing object after the enclosing object is instantiated. Once included in the enclosing object, a reference to an interface of the enclosed object may be returned in response to an external request to access the interface (see Leach et al., Abstract, col. 11, lines 23-34 and col. 21, line 65 to col. 22, line 21).

In contrast, claim 1 recites a combination of steps including, among other things, dispatching a request to an object to facilitate processing of a method of an interface specified at runtime, and returning a result of the processed method by the object.

The Examiner asserts that Leach et al. "discloses dispatching the request to an object to facilitate processing of the method of the interface (Fig. 7C, and requests received by an enclosed object are passed to the enclosing object, line 67 column 22 to

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

line 1 column 23)" (see Final Office Action, page 6, lines 14-20). Applicants respectfully disagree for the following reasons.

First, as explained by Applicants in the response filed October 7, 2002, the block diagram illustrated in Fig. 7C of Leach et al. (referenced by the Examiner) does not show the dispatching of a request to an object to facilitate processing of the method of an interface. Instead, this block diagram shows a multitype object ("MTO") after adding an interface using particular method (e.g., AddObject). The IUnknown interface included in the MTO is capable of providing a pointer to an external interface and/or other MTO interfaces (see Leach et al., col. 24, lines 27-44).

Second, the ability of Leach et al. to pass requests received by an enclosed object to an enclosing object (cited by the Examiner) is associated with static aggregation. As explained, static aggregation requires that an enclosing object have advance (i.e., before runtime) knowledge of the interfaces it wishes to aggregate. Claim 1 includes the step of generating at runtime a class that implements an interface specified at runtime having a method. Accordingly, the static aggregation process of passing of requests to an enclosing object discussed by Leach et al. in col. 22, line 67 to col. 23, line 3 cannot teach the dispatching step of claim 1 because this step includes dispatching the request to an object to facilitate processing of a method of the interface that is specified at runtime.

Third, the dynamic aggregation processes disclosed by Leach et al. in col. 23, lines 4-23, also fail to teach the dispatching step of claim 1. According to Leach et al., the enclosing object "provides a method for registering instantiated interfaces and for later retrieving references to them." (see Leach et al., col. 23, lines 5-7). Therefore,

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

Leach et al. requires an enclosing object method to register interfaces as they are instantiated and only provides references to the interfaces. Leach et al. does not teach processing the method of the interface, much less dispatching a request to an object to facilitate the processing of the method. Instead, Leach et al. teaches an enclosing object (e.g., multitype object) that uses a method to retrieve references to registered interfaces. Accordingly, Leach et al. cannot teach the dispatching step of claim 1 because the reference does not disclose dispatching a request to an object to facilitate processing of a method of an interface. The method invoked by the enclosing object as taught by Leach et al. is not an object. Further, the enclosing object taught by Leach et al. cannot be equivalent to the object of claim 1 that receives the dispatched request because the enclosing object performs aggregation services which are not associated with the processing of any methods of interfaces implemented by the enclosing object. The aggregation services merely add objects and interface lists to an enclosing object to allow the enclosing object to retrieve pointers to interfaces implemented by the enclosing object.

Additionally, Applicants traverse the Examiner's interpretation of Leach et al. in view of the combination of steps recited in claim 1. For example, the Examiner asserts that Leach et al. teaches generating at runtime a class that implements an interface specified at runtime having a method and creating an instance of the class (see Final Office Action, page 2, lines 19-21). In particular, the Examiner asserts that the "objects" disclosed in col. 11, lines 15 and 21-22 of Leach et al. are the same as the class instance created in claim 1. Applicants disagree. The objects created by Leach et al. are enclosed objects which are objects or interfaces added to an enclosing object. The

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

enclosed objects are those objects or interfaces implemented during runtime and are not the same as the class instance as specified in claim 1.

The Examiner also asserts that Leach et al. teaches returning a result of a processed method as recited in claim 1. The Examiner contends that Applicants are "arguing [a] limitation that is disclosed in the specification but not claimed before." Applicants respectfully disagree.

Claim 1 was amended in the October 7, 2002 response to include the step of returning a result of the processed method by the object. Recitations that are similar (albeit not identical) to this added step were recited in claim 3, which was canceled in the October 7, 2002 response. Accordingly, Applicants traverse the Examiner's assertions that Applicants are arguing recitations not previously claimed, and request the Examiner properly address the arguments presented below corresponding to the returning step recited in claim 1.

Leach et al. does not teach returning a result of the processed method by the object, as recited in claim 1. As admitted by the Examiner, the QueryInterface method implemented by Leach et al. returns pointers to exposed interfaces (see Final Office Action, page 3, lines 6-8). Returning pointers is not the same as returning a result of a processed method of the interface specified at runtime, as recited in claim 1. In fact, Leach et al. does not even disclose processing a method of an interface to return a result. Instead, Leach et al. processes a method in an enclosing object (e.g., multitype object) that returns references to interfaces added to the enclosing object (See Leach et al., col. 23, lines 5-7). Accordingly, Leach et al. cannot teach the returning step of claim 1 because the reference does not teach processing a method of an interface enclosed

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

in the enclosing object, but instead processes a single method of the enclosing object to return a pointer to an enclosed interface.

Because Leach et al. fails to teach every recitation of claim 1, Applicants respectfully request that the rejection of this claim under 35 U.S.C. § 102(e) be withdrawn and the claim allowed.

Claims 2 and 4 depend from claim 1. As explained, claim 1 is distinguishable from Leach et al. Accordingly, claims 2 and 4 are also distinguishable from this reference for at least the same reason set forth for claim 1 and Applicants request that the rejection of these claims under 35 U.S.C. § 102 be withdrawn and the claims allowed.

The Examiner also alleges that Leach et al. teaches specifying an object to process method invocations on the class instance, as recited in claim 4. Applicants disagree for the following reasons.

First, the process of determining which interface to retrieve and how to invoke the interface, as taught by Leach et al. is not the same as the specifying step recited in claim 4. As explained in the October 7, 2002 response, Leach et al. provides a method for modifying a determination of which interfaces to retrieve and how to invoke them "in combination if more than one instance of the same interface is present in the aggregate object" (i.e., enclosing object) (see Leach et al., col. 23, lines 9-12). This process taught by Leach et al. is not equivalent to specifying an object to process method invocations on the instance, as recited in claim 4 because, at the very least, there is no specification of an object that processes method invocations. Moreover, the method provided by Leach et al. is limited to information on how to invoke interfaces in combination if more

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

than one instance of the same interface is present. This limitation has no relationship to the recitations of claim 4.

The Examiner responds to these arguments by simply reiterating the same position held in the previous Office Action, that the disclosure in col. 23, lines 9-10 of Leach et al. teaches the specifying step of claim 4. Accordingly, the Examiner has not presented evidence in light of Applicants' arguments that Leach et al. anticipates claim 4. Because the Examiner has not shown that Leach et al. teaches the recitations of claim 4, Applicants submit that the rejection of this claim under 35 U.S.C. § 102(e) is improper and should be withdrawn.

Second, the query function member that is invoked by Leach et al. merely retrieves a reference to an aggregated interface, and thus cannot teach the step of specifying an object to process method invocations on the instance, as recited in claim 4. The Examiner appears to take the position that the query function member of the multitype object taught by Leach et al. is the same as the object specified in claim 4. Applicants disagree. The query function member is not an object and there is no evidence in Leach et al. that shows the function member is even specified when the multitype object is created. Therefore, the invocation of a query function member of the multitype object taught by Leach et al. is not the same as the specifying step of claim 4.

Accordingly, for at least the above reasons, Applicants respectfully request that the rejection of claim 4 under 35 U.S.C. § 102(e) be withdrawn and the claim allowed.

Claims 6 and 12 include recitations similar to those of claim 1. As explained, claim 1 is distinguishable from Leach et al. Accordingly, claims 6 and 12 are also distinguishable from this reference for at least the same reason set forth in connection

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

with claim 1 and Applicants request that the rejection of these claims under 35 U.S.C. § 102(e) be withdrawn and the claims allowed.

Claims 7 and 9 depend from claim 6. As explained, claim 6 is distinguishable from Leach et al. Accordingly, claims 7 and 9 are also distinguishable from this reference for at least the same reason set forth in connection with claim 1. Further, claim 9 includes recitations similar to those of claim 4. Accordingly, claim 9 is also distinguishable from Leach et al. for at least the same reasons set forth in connection with claim 4. Based on the foregoing, Applicants request that the rejection of claims 7 and 9 under 35 U.S.C. § 102(e) be withdrawn and the claims allowed.

Applicants traverse the rejection of claims 5 and 10 under 35 U.S.C. § 103(a) because a prima facie case of obviousness has not been made by the Examiner. To establish a prima facie case of obviousness under 35 U.S.C. § 103(a), each of three requirements must be met. First, the reference or references, taken alone or combined, must teach or suggest each and every element recited in the claims (See M.P.E.P. § 2143.03 (8th ed. 2001).) Second, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art to combine the references in a manner resulting in the claimed invention. Third, a reasonable expectation of success must exist. Moreover, each of these requirements must "be found in the prior art, and not be based on applicant's disclosure" (See M.P.E.P. § 2143 (8th ed. 2001)).

In rejecting claims 5 and 10, the Examiner admits that Leach et al. does not teach an invocation handler. To supplement this missing recitation, the Examiner asserts that the process execution handler taught by Hailpern et al. is equivalent to an

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

invocation handler and that it would have been obvious to apply the execution handler to the system of Leach et al." because this allows dynamic aggregating objects" (see Final Office Action, page 4, lines 16-20).

In the response filed October 7, 2002, Applicants presented arguments distinguishing claims 5 and 10 from Leach et al. and Hailpern et al. and presented arguments why the Examiner failed to establish a prima facie case of obviousness. In response, the Examiner merely addressed one of the arguments presented by Applicants by merely stating that "while this may be true [the handler of Hailpern et al. is not in the same context as the handler claimed] it does not preclude using Hailpern in the claim rejections." The Examiner states nothing else regarding the issues involved with the rejections of claims 5 and 10 under 35 U.S.C. § 103(a). Applicants disagree with the Examiner for the following reasons.

First, the fact that Hailpern et al. does not teach or suggest (as Admitted by the Examiner) a handler that is in the same context as the invocation handler of claims 5 and 10 does preclude using the reference to reject these claims under 35 U.S.C. § 103(a). As explained above, the Examiner has a burden to show (1) that Hailpern et al. and Leach et al., taken alone or combined, teaches or suggests each and every element recited claims 5 and 10 (See M.P.E.P. § 2143.03 (8th ed. 2001).), (2) that there is some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art to combine the Hailpern et al. and Leach et al. in a manner resulting in the inventions recited in claims 5 and 10, and (3) that there is a reasonable expectation of success in making the combination.

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

The Examiner has failed to meet any of the above listed burdens in rejecting claims 5 and 10. As explained in the response filed October 7, 2002, the process execution handler taught by Hailpern et al. carries out "necessary processing" for a server that performs virus checking in a server network environment (see Hailpern et al., col. 15, lines 48-40). The execution handler is not an invocation handler that receives method invocations, as recited in claims 5 and 10.

Also, there is no reasonable expectation of success in implementing a virus scanner exception handler, as taught by Hailpern et al., in the method invocation processes taught by Leach et al. One of ordinary skill in the art would have appreciated, at the time of the invention, that such a combination is not suggested in the references themselves in any possible interpretation and that the asserted combination is not feasible. There is no suggestion in either reference that would motivate one of ordinary skill in the art to use virus exception handlers in the system for aggregating objects taught by Leach et al.

Further, when given the opportunity to elaborate on the rejection of claims 5 and 10, the Examiner merely concludes that Hailpern et al. may be used to reject these claims without presenting evidence to support the conclusion. Applicants submit this is improper and request reconsideration of the rejection of claims 5 and 10 under 35 U.S.C. § 103(a).

The Examiner also maintains the position that Leach et al. teaches generating at runtime a class that implements the interface by generating code for each of the methods included in the interface, as recited in claims 5 and 10 (see Final Office Action, page 6, lines 14-21). Applicants disagree.

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

As explained in the response filed October 7, 2002, the code table 3 taught by Leach et al., referred to by the Examiner, is a listing of pseudo code for a **class definition of an object** enclosed in an aggregate object (see Leach et al., col. 14, lines 23-28). There is no mention or suggestion in Leach et al., particularly in code table 3, of generating a class at runtime that implements an interface by generating code for each of the methods in the interface, as recited in claims 5 and 10. Further, merely presenting a class definition does not show **generating at runtime** a class that implements an interface by generating code for each of a plurality of methods of an interface indicated at runtime. In fact the class definition presented in Code Table 3 of Leach et al. is programmed before runtime. Accordingly, Leach et al. cannot teach or suggest the generating step recited in claims 5 and 10. Also, Hailpern et al. fails to make up for the deficiencies of Leach et al.

Based on the foregoing, Applicants request that the rejection of claims 5 and 10 under 35 U.S.C. § 103(a) be withdrawn and the claims allowed.

Claims 11 and 13 include recitations similar to those of claims 1 and 5. As explained, claims 1 and 5 are distinguishable from Leach et al. and Hailpern et al. Accordingly, claims 11 and 13 are also distinguishable from these references for at least the same reason set forth in connection with claims 1 and 5. Further, claim 11 is also distinguishable from Hughes because that reference does not teach or suggest, alone or in combination with Leach et al. and Hailpern et al., at least dispatching a request to an invocation handler object and returning a value from the handler object to a proxy class instance, as recited in the claim 11.

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

Further, the Examiner maintains the position that Hughes teaches a proxy class generated at runtime, as recited in claims 11 and 13. The Examiner reaches this conclusion by once again reiterating the arguments presented in the previous Office Action without providing evidence or an explanation to support the conclusion. Accordingly, Applicants reiterate the arguments presented in the October 7, 2002 response and request that the Examiner address them accordingly. As explained, the proxy class disclosed by Hughes is not generated at runtime. Instead, a manufacturer generates code for interfaces, and other functionalities associated with the proxy class (see Hughes, col. 4, lines 18-31 and 60-67. As disclosed by Hughes, a proxy class is not generated at runtime, but rather **a user dynamically specifies at runtime** the "customization" of an instance of the proxy class (see Hughes, col. 8, lines 6-8).

Accordingly, Hughes, Leach et al. and Hailpern et al., alone or in combination, fail to teach or suggest the recitations of claims 11 and 13 and Applicants request that the rejection of these claims under 35 U.S.C. § 103(a) be withdrawn and the claims allowed.

Applicants respectfully request that this response under 37 C.F.R. § 1.116 be entered by the Examiner, placing claims 1, 2, 4-7, and 9-13 in condition for allowance. Applicants respectfully point out that the final action by the Examiner presented some new arguments as to the application of the art against Applicant's invention. Further, the Examiner did not response to all of Applicants' earlier positions presented in the October 7, 2002 response. Accordingly, it is respectfully submitted that the entering of this response would allow Applicants to reply to the final rejections and place the application in condition for allowance.

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com

Finally, Applicants submit that the entry of this response would place the application in better form for appeal, should the Examiner dispute the patentability of the pending claims.

In view of the foregoing amendments and remarks, Applicants respectfully request the reconsideration and reexamination of this application and the timely allowance of claims 1, 2, 4-7, and 9-13.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

Dated: February 11, 2003

By: _____
Joseph E. Palys
Reg. No. 46,508

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

1300 I Street, NW
Washington, DC 20005
202.408.4000
Fax 202.408.4400
www.finnegan.com